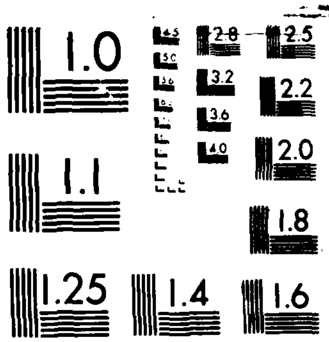
AD-A160 247 NETWORK AGGREGATION CONCEPTS(U) GEORGIA INST OF TECH 1/1
ATLANTA PRODUCTION AND DISTRIBUTION RESEARCH CENTER
A V IYER ET AL. MAY 85 PDRC-85-04 N00014-83-K-0147
UNCLASSIFIED F/G 15/5 NL



MICROCOPY RESOLUTION TEST CHART

AD-A168 247

Georgia Institute of Technology

NETWORK AGGREGATION CONCEPTS

by

A. V. Iyer
John J. Jarvis
H. Donald Ratliff

PDRC 85-04

PRODUCTION and DISTRIBUTION RESEARCH CENTER

SCHOOL OF INDUSTRIAL
AND SYSTEMS
ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY
A UNIT OF THE UNIVERSITY SYSTEM
OF GEORGIA
ATLANTA, GEORGIA 30332

PDRC Report Series 85-04
May 1985

(2)

NETWORK AGGREGATION CONCEPTS

by

A. V. Iyer
John J. Jarvis
H. Donald Ratliff

PDRC 85-04

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

This work was supported by the Office of Naval Research under Contract No. N00014-83-K-0147 and the Joint Deployment Agency under contract N00014-84-C-0578. Reproduction is permitted in whole or in part for any purpose of the U. S. Government.

1.0 INTRODUCTION

This report considers the use of aggregation in modeling and solving network based logistics planning problems. Aggregation combines, or pools, certain attributes of a problem to facilitate its modeling and solution. Frequently, after the model is solved it is necessary to disaggregate the pooled attributes to their original level of detail. This allows determination of the acceptability of the model output in making the suggested decisions. If the aggregate model yields an unacceptable solution, a mechanism is required by which the solution can be modified to make it acceptable.

Aggregation imposes a natural decision hierarchy with higher level aggregate decisions restricting lower level, more detailed, decisions. For large scale planning situations requiring involvement of a number of participants with skills in different areas, this can be very important. A structure is required to coordinate the efforts of the different participants. This structure should

- provide initial guidelines for aggregation,
- utilize detailed plans for revising the aggregate plan,
- iterate until an acceptable overall plan is obtained.

A mathematical structure for the aggregation process will be developed which allows a large problem to be solved as a sequence of smaller problems. This structure can also serve as a coordinating instrument for the decision making hierarchy.

Motivation for examining aggregation in the context of network based planning problems is summarized as follows. Detailed data is often not available at the beginning of the planning process, but aggregate data can be estimated. In such cases solving the aggregate problem frequently

indicates what detailed data is required before a solution can be implemented.

Aggregating data tends to reduce variation. For example, monthly demand is likely to have less variation than daily demand. Consideration of a problem at a detailed level often requires that parameters be modeled as random variables, while an aggregate level allows the parameter to be considered as essentially deterministic. This significantly decreases model complexity.

In many situations a detailed planning model is too large to be solved with available computer resources and planning time. Further, in many planning environments the objective is not merely to optimize a given function but to generate a plan which is satisfactory with respect to multiple objectives, some of which are not quantitfiable. All or parts of the model may be solved more than once, with the user interacting to guide the process to an acceptable solution. This implies a need for reorganizing a large monolithic model into a set of smaller models which when linked together represent the original model. Aggregation can be used to devise such a decomposition strategy.

Aggregation concepts developed here are motivated by the need for aggregation in the SCOPE system developed in PDRC Report 84-09. One of the fundamental modeling components of the SCOPE system is the assignment of movement requirements to transportation channels given the MR assignment problem. The optimization model used is a transportation model. This model will be referred to throughout this report. It is discussed in detail in PDRC Report 84-09. The generalized version of this model is currently being coded as the MRMATE model in the MODES system under developed for the Joint Deployment Agency. The methodology

developed here is applicable to both the pure and the generalized forms of the model in that report. To simplify the presentation here, standard notation will be used rather than the more complex notation required to describe the movement requirement assignment model.

2.0 AGGREGATION

Consider the example transportation problem given below and its corresponding network representation given in Figure 2.1.

$$\text{Min} \quad \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} x_{ij}$$

Subject to:

$$\begin{array}{lllllllllll}
(1) & x_{14} + x_{15} + x_{16} & & & & & & & = a_1 & & \\
(2) & & + x_{25} + x_{27} & & & & & & = a_2 & & \\
(3) & & & + x_{34} + x_{36} + x_{37} & & & & & = a_3 & & \\
(4) & x_{14} & & + x_{34} & & & & & = a_4 & & (\mathrm{I}) \\
5) & \quad x_{15} & + x_{25} & & & & & & = a_5 & & \\
(6) & \quad\quad x_{16} & & + x_{36} & & & & & = a_6 & & \\
(7) & & + x_{27} & + x_{37} & & & & & = a_7 & &
\end{array}$$

$$x_{ij} \geq 0 \quad \text{for } i = 1..3,\ j = 1..4$$

In the model, x_{ij} is the flow on the arc from source node i to sink node j and c_{ij} is the corresponding cost. Nodes on the left-hand side of the transportation network are source nodes and nodes on the right-hand side are sink nodes.

This report concentrates only on aggregation of nodes on the right-hand side of the transportation network (i.e. sink nodes only). This same logic can be extended to aggregation of both sources and sinks by performing the aggregation in two steps.

Consider aggregation of nodes 4 and 5 in Figure 2.1 to form an aggregate node 1 in Figure 2.2, and nodes 6 and 7 in Figure 2.1 to form

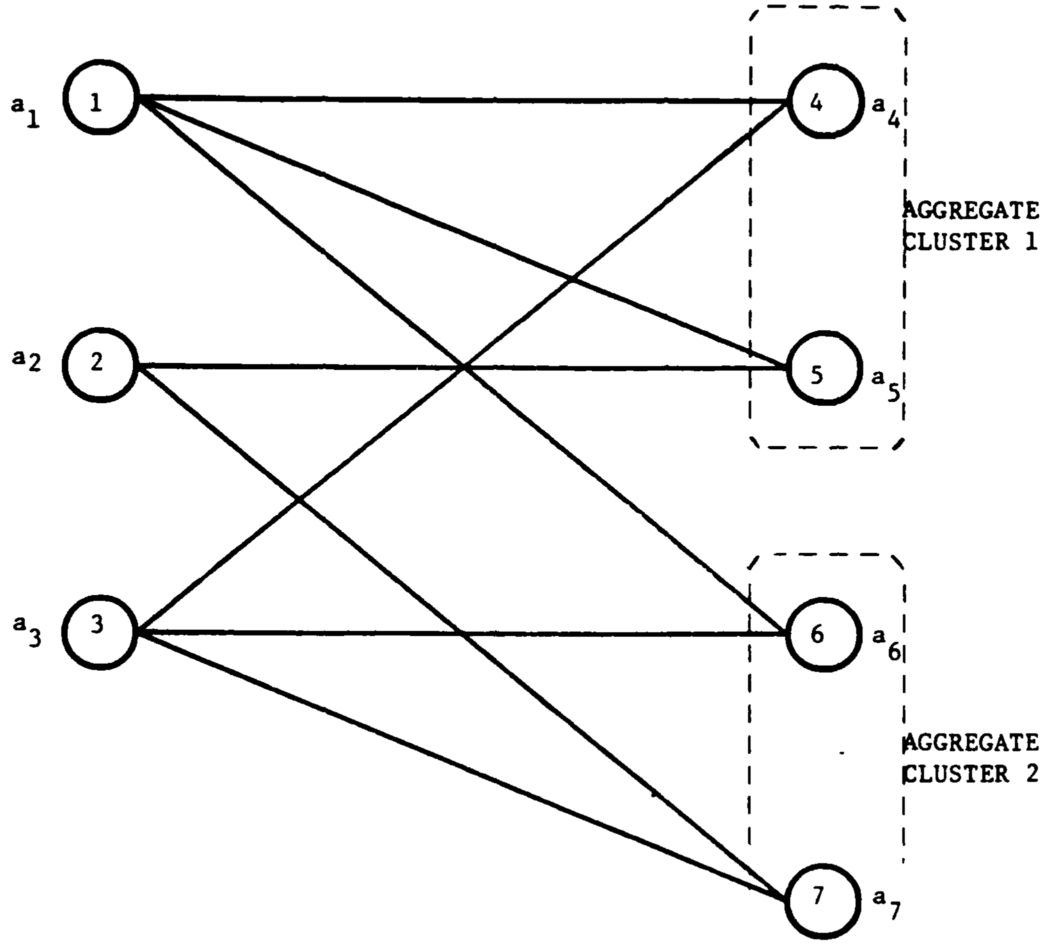


Figure 2.1. Detailed Network Example.

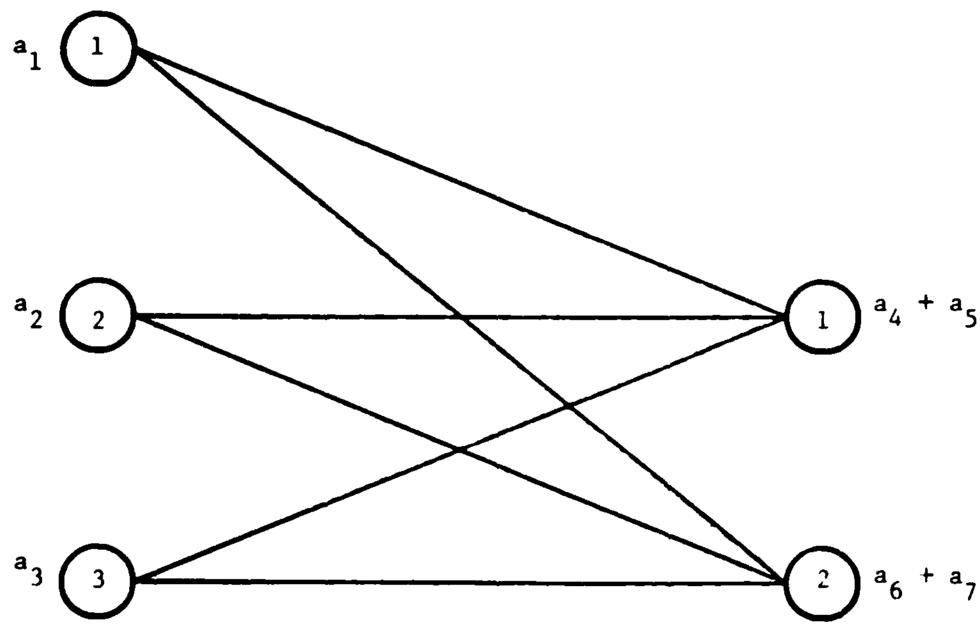


Figure 2.2. Aggregate Network Example.

aggregate node 2 in Figure 2.2. Constraints corresponding to the network in Figure 2.2 are shown in (II) below.

$$
\begin{array}{llr}
(8) & y_1^1 + y_1^2 = a_1 & \\
(9) & y_2^1 + y_2^2 = a_2 & \\
(10) & y_3^1 + y_3^2 = a_3 & \text{(II)} \\
(11) & y_1^1 + y_2^1 + y_3^1 = a_4 + a_5 & \\
(12) & y_1^2 + y_2^2 + y_3^2 = a_6 + a_7 &
\end{array}
$$

$$y_i^r \geq 0 \quad \text{for } i = 1 \ldots 3,\ r = 1 \ldots 2$$

In the model, y_i^r represents the flow on the arc from source node i to aggregate sink node r and c_i^r is the corresponding cost per unit of flow.

Examining the information that is lost in the aggregate problem in Figure 2.2 the following observations can be made. There is a single cost on each aggregate arc in Figure 2.2. This implies that flow on the aggregate arc (1,1) incurs the same cost as it would on either of the detailed arcs (1,4) or (1,5). If the two detailed arc costs are not the same, then some amount of error is introduced by combining the two costs into a single cost. If the amount of flow on each of these arcs is known in advance, the aggregate cost could be weighted so that it would

accurately reflect the detailed cost. While the detailed flows are not known in advance, there is the potential for iteratively improving the weighted aggregate cost after solving an aggregate problem and then disaggregating.

A simple method for specifying the capacity of aggregate nodes is to set the aggregate capacity equal to the sum of the corresponding detailed node capacities. However, such a capacity may not be meaningful when the nodes model different entities. For example, aggregating sea and air channel capabilities in the MR assignment model allows materiel to be sent by either air or sea. If some materiel can only go by sea then the aggregate node capacity exceeds the practical total which can be shipped.

There is also a problem in specifying the capacity of aggregate arcs. For example, consider the aggregate arcs (1,1) and (2,2) in Figure 2.2. Since (1,1) represents both arcs (1,4) and (1,5) in Figure 2.1, it can have a maximum flow of $(a_4 + a_5)$. However (2,1) in Figure 2.2 represents only (2,5) in Figure 2.1. Hence, it can only have a maximum flow of a_5. Procedures must be developed for passing this information to the aggregate problem while retaining its transportation structure. Simply putting the sum of the associated node capacities as capacities on the aggregated arcs will not suffice.

This report outlines approaches to deal with each the above issues. It also proposes a format which can be used to examine alternative formulations of the aggregate problem to determine an optimum aggregation.

2.1 General Aggregation Model

This section describes approaches to aggregation methods presented in the literature. The formulations are discussed in Zipkin [5] and Taylor [4].

When the aggregate problem is formulated, the following parameters must be specified

- capacities on the aggregate sink nodes
- costs on the aggregate arcs
- capacities (upper limits on flow) on the aggregate arcs.

The detailed problem is

$$\text{Min} \quad \sum_i \sum_j c_{ij} x_{ij}$$

$$\text{subject to:} \quad \sum_j x_{ij} = a_i \qquad \text{for } i = 1,2...M$$

$$\sum_i x_{ij} = b_j \qquad \text{for } j = 1,2...N$$

$$x_{ij} \geq 0$$

Construction of the aggregate problem is essentially a two step procedure. First, the nodes to be aggregated (e.g. Figure 2.3) are combined into a single aggregate node (e.g., Figure 2.4). This results in a set of parallel arcs as indicated in Figure 2.4. These parallel arcs are then aggregated into a single aggregate arc.

The aggregate node capacity is derived from the capacities of the individual nodes forming the cluster. If a linear relationship is assumed (as a simplifying assumption), then the aggregate capacity can be expressed as

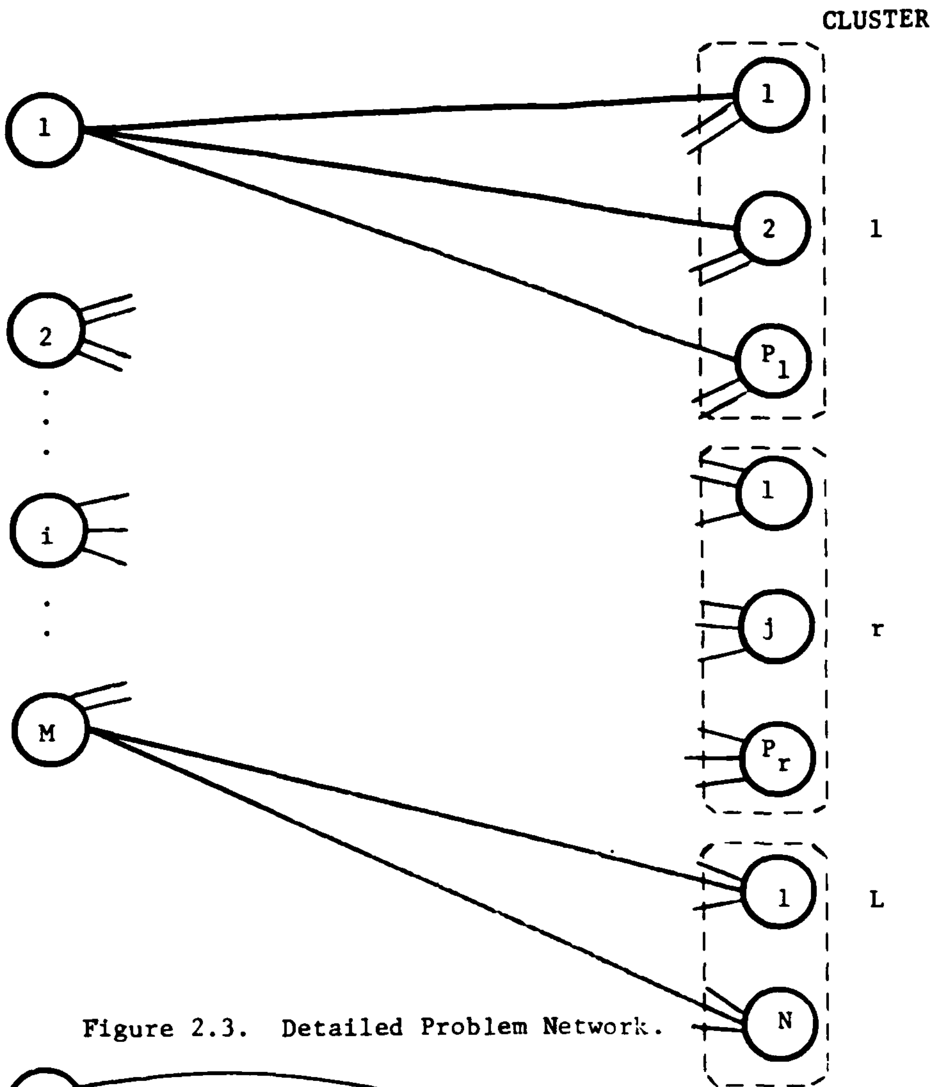


Figure 2.3. Detailed Problem Network.

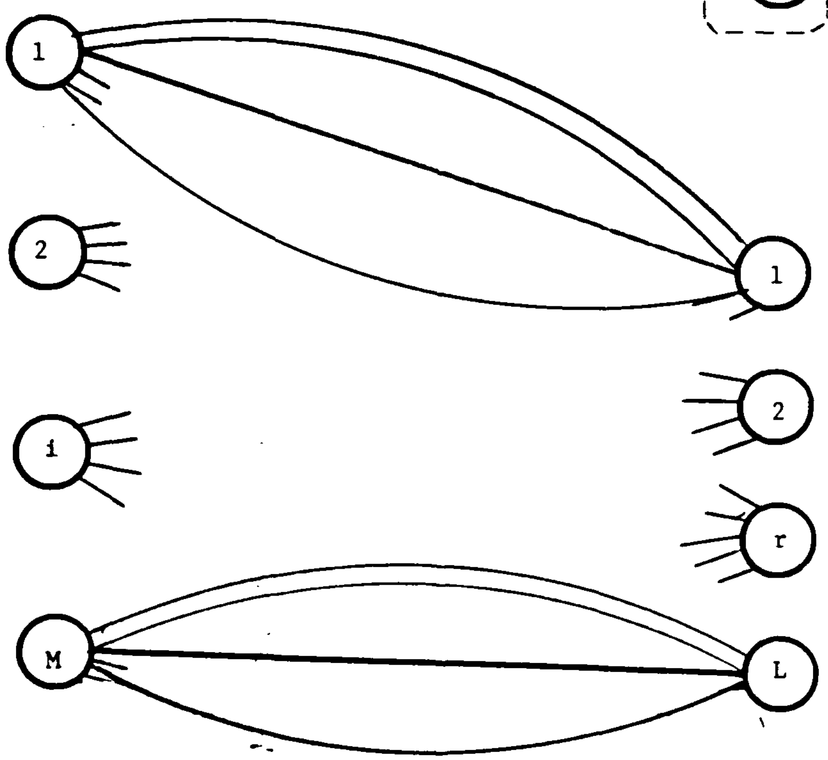


Figure 2.4. Network in Figure 2.3 After Node Aggregation.

$$b_r = \sum_{j \epsilon P_r} t_j \, b_j$$

where P_r is the set of nodes in the rth aggregation cluster, b_j is the capacity of node j in P_r for the detailed problem, and t_j is a user defined multiplier between zero and one.

The motivation for different choices of t_j are as follows. If there are arcs from all source nodes to all sink nodes in P_r, (e.g., any movement requirement can be sent via any channel) then it is appealing to represent the aggregate node capacities as the sum of the individual node capacities. This implies that $t_j = 1$.

The motivation for a choice of t_j less than 1, is to prevent the aggregate capacity from over estimating the actual capacity of the system. For example, if some movement requirements can only move as outsize cargo, an aggregate node representing both oversize and outsize cargo would overestimate capability if it were simply the sum of the individual detailed oversize and outsize channel capabilities. The amount of usable capability can be controlled by the factor t_j.

When node aggregation is performed, information is lost regarding the detailed capacities on nodes to which each of the multiple arcs are incident. The problem, as shown in Figure 2.5 represents the aggregate model after the parallel arcs in Figure 2.4 have been aggregated.

If a relationship between the detailed arc flows x_{ij} and the aggregate arc flow y_i^r is assumed, then disaggregation is greatly simplified. A simple relation is

$$x_{ij} = s_{ij} * y_i^r \quad \text{where } j \epsilon P_r$$

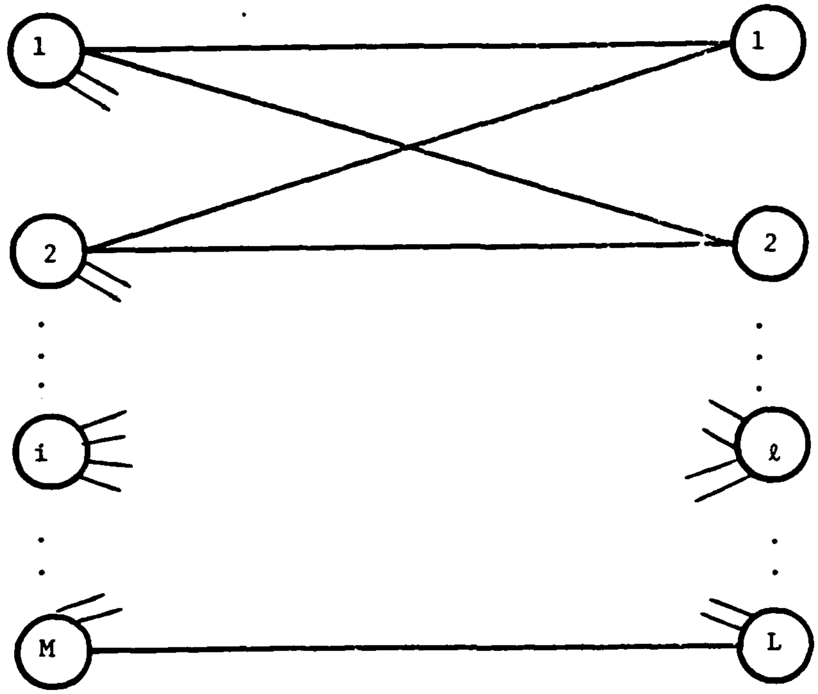


Figure 2.5. Network in Figure 2.4 After Parallel Arc Aggregation.

where s_{ij} is the fraction of the aggregate arc flow which is to be allocated to the detailed arc ij.

Such a relationship requires that the values of s_{ij} be specified. For a given t_j, if s_{ij} is assumed to be a function of the data regarding sink nodes alone, (i.e. independent of the source node data) then the condition that x_{ij} be a feasible solution to the detailed problem permits the determination of the appropriate value for s_{ij}.

From the detailed problem, the aggregate problem is formulated corresponding to the network shown in Figure 2.5. The aggregate node capacities are as defined earlier and y_i^r refer to the flows on the aggregate arcs (i,r).

Aggregate Problem.

$$\sum_{i=1}^{M} \sum_{r=1}^{L} c_{ir} y_i^r$$

$$\sum_i y_i^r = b_r = \sum_{j \in P_r} t_j b_j$$

$$\sum_r y_i^r = a_i$$

$$y_i^r \geq 0$$

If $t_j = 1$ is selected, and

$$x_{ij} = s_{ij} y_i^r$$

then, in order to ensure that the x_{ij} obtained from the solution to the aggregate problem is feasible to the original problem, the feasibility conditions must be checked as follows

$$\sum_j x_{ij} = \sum_r \sum_{j \epsilon P_r} s_{ij} y_i^r$$

$$= \sum_r y_i^r \sum_{j \epsilon P_r} s_{ij}$$

$$= a_i \sum_{j \epsilon P_r} s_{ij}$$

$$= a_i \quad \text{if} \quad \sum_{j \epsilon P_r} s_{ij} = 1$$

From the feasibility conditions

$$\sum_{j \epsilon P_r} s_{ij} = 1.$$

Next, the conservation of flow conditions are checked at the sink nodes

$$\sum_i x_{ij} = \sum_i s_{ij} y_i^r .$$

Assuming that s_{ij} depends on the sink data alone then

$$\sum_i s_{ij} y_i^r = s_{ij} \sum_i y_i^r$$

$$= s_{ij} \sum_{k \epsilon P_r} b_k$$

$$= b_j .$$

The above relation holds for

$$s_{ij} = b_j / (\sum_{k \epsilon P_r} b_k)$$

If s_{ij} is independent of the source nodes and $t_j = 1$ then

$$s_{ij} = b_j / (\sum_{j \epsilon P_r} b_j)$$

Also, for this specification of t_j and s_{ij} values, we have the objective function value of the feasible solution generated to the original problem equal to the objective function value of the aggregate problem, if the aggregate arc costs are

$$c_{ir} = \sum_{j \epsilon P_r} c_{ij}\, s_{ij}$$

This follows since the objective function of the detailed problem is

$$\begin{aligned} \sum_i \sum_j c_{ij}\, x_{ij} &= \sum_i \sum_r \sum_{j \epsilon P_r} c_{ij}\, s_{ij}\, u_i^r \\ &= \sum_i \sum_r \sum_{j \epsilon P_r} (s_{ij}\, c_{ij})\, y_i^r \\ &= \sum_i \sum_r c_{ir}\, y_i^r \end{aligned}$$

for the cost of the aggregate arc as specified above.

If the values of the node capacities involved in an aggregate node are equal and $t_j = 1$, then $s_{ij} = 1/(\ P_r)$ for j in P_r. This essentially means that the arcs flows are distributed equally across all the incident detailed arcs.

The discussion in this section permits the use of a detailed problem (assuming that there is an arc from every source to every single node) together with a specified aggregation of nodes in order to define an aggregate problem with aggregate costs

$$c_{ir} = \sum_{j \epsilon P_r} c_{ij} s_{ij}$$

where

$$s_{ij} = \frac{b_j}{\sum_{k \epsilon P_r} b_k} \quad \text{for } j \epsilon P_r.$$

and aggregate node capacities

$$b_r = \sum_{k \epsilon P_r} b_k.$$

Once the aggregate problem is solved, a disaggregation scheme is available which sets

$$x_{ij} = s_{ij} y_i^r.$$

This always provides a feasible solution to the original problem.

Unfortunately, in many deployment planning situations, the transportation problem is sparse with the sparsity reflecting feasible allocations of movements to transportation modes (i.e. some movement requirements cannot be assigned to certain channels). In such cases there is not an arc from every source node to every sink node and the disaggregation outlined in this section will not always provide a feasible solution.

The solution to the detailed problem generated is guaranteed to be feasible only if all such nodes in a cluster have arcs to the same set of source nodes. Since a specified aggregation may not satisfy this condition, formulations need to be examined which generate feasible solutions when sink nodes with arcs to different source nodes are aggregated. Such a procedure will be provided in Section 5.0.

3.0 DISAGGREGATION ISSUES:

Given a detailed problem, an aggregate model can be constructed as illustrated in the previous section. Once the aggregate model is solved, the next question is how to disaggregate the aggregate solution. Aggregation is used to determine a gross allocation of resources to the aggregate sink nodes. Disaggregation distributes the allocated aggregate resources across the detailed nodes and arcs.

Aggregation is usually motivated by a desire to decompose a large problem into smaller manageable subproblems. A desirable feature of any disaggregation is that it permit the subproblems to be solved independently. This permits the subproblem solution process to occur in parallel. The objective of disaggregation is usually to generate "good feasible" solutions to the detailed problem. It is also desirable to be able to evaluate alternative solutions to the detailed problems. The characteristics of a good disaggregation procedure are that it:

(1) enable parallel processing of the clusters, so that the time is merely the largest time among all clusters. This is useful from the perspective of use of aggregation as a co-ordinating procedure.

(2) generate a feasible solution to the original problem. This permits the procedure to be interrupted by the user interface so as to provide input to speed up the overall problem solution. Also, even though optimality of the entire problem, may not have been reached the solution at the end of each iteration offers suboptimal scenarios which may be satisfactory

from the perspective of satisfying certain other qualitative considerations.

Disaggregation procedures reported in the literature are essentially of two types: fixed weight disaggregation and optimal disaggregation. Fixed weight disaggregation uses the same weights used in the aggregated problem formulation. It is fixed in the sense that it is a single, noniterative multiplication procedure. It uses the s_{ij} values to generate a feasible solution to the detailed problem as follows.

$$x_{ij} = s_{ij} * y_i^r \quad \text{for } j \varepsilon P_r$$

This provides a feasible solution to the detailed problem provided that there is an arc from every source node to every sink node. This procedure is easy, quick, and does not involve any specific algorithmic procedures. However, since the s_{ij} values were assumed to be independent of the source nodes, the solution generated for sparse problems may be infeasible because information regarding the missing arcs is not utilized.

Optimal disaggregation is optimal in the sense that, given a partition of the source flow allocation to the clusters, this flow is routed optimally within each cluster. The solution generated is no worse then the fixed disaggregation case.

The optimal disaggregation procedure allows each cluster of nodes to be evaluated independently from the aggregate problem solution. This permits parallel solution of the disaggregation problems for each cluster. The essential idea is to flow the aggregate flows back

through the detailed arcs and nodes. The problem constructed for each cluster would be a transportation problem. It would have a number of source nodes equal to the number of aggregate basic feasible arcs incident to the aggregate cluster node, and a number of sink nodes equal to the number of detailed nodes aggregated into the cluster. The costs on the arcs are the costs on the detailed problem, with the aggregate flows forming the capacities on the source nodes and capacities (obtained from the detailed problem) on the sink nodes. The disaggregation problem for each cluster r is as follows

$$\begin{aligned} \text{Min} \quad & \sum_i \sum_j c_{ij}\, x_{ij} \\ \text{subject to:} \quad & \sum_{j \varepsilon P_r} x_{ij} = y_i^r \quad \text{for } i = 1,2,\ldots M \\ & \sum_i x_{ij} = b_j \quad \text{for } j \varepsilon P_r \\ & x_{ij} \geq 0 \end{aligned}$$

The corresponding general network model is shown in Figure 3.1. Since the disaggregation problems allow flows only on arcs which exist in the original problem, a feasible solution, if one exists is assured to the detailed problem.

For the example whose detailed formulation is given in (I) and whose aggregate formulation is given in (II), disaggregation involves allocating the aggregate flow variables y_i^r to the detailed flow variables x_{ij}. Since there are two aggregate clusters, there are two disagregation problems.

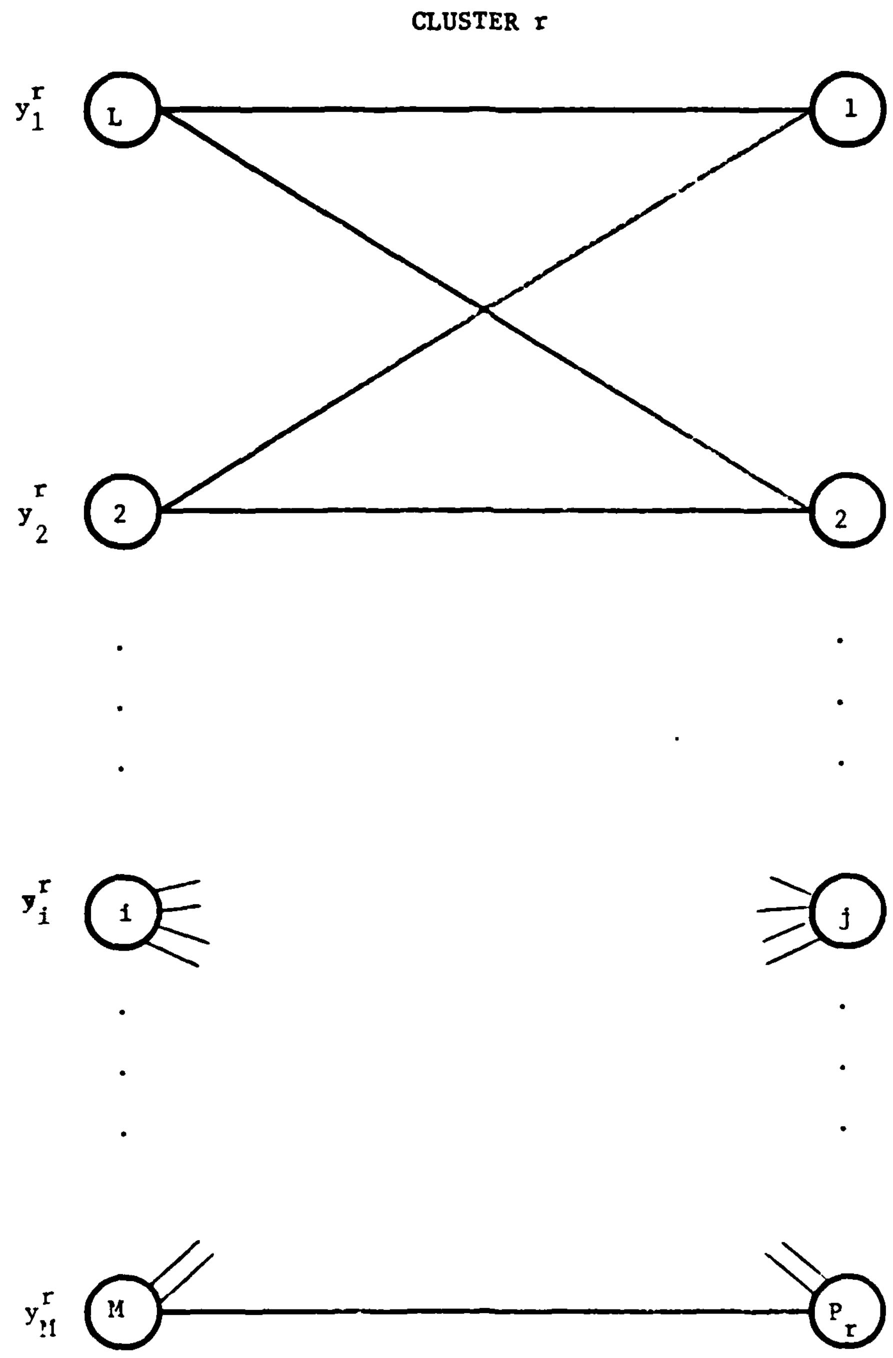


Figure 3.1. Optimal Disaggregation Network for Cluster r.

For each cluster, the following is the optimal disaggregation problem. The values of y_i^r are specified from the solution to the aggregate problem. The networks are shown in Figure 3.2 for cluster 1 and in Figure 3.3 for cluster 2. The constraints corresponding to these networks are given in (III) and (IV) respectively.

$$
\left.\begin{array}{llllll}
(13) & x_{14} + x_{15} & & & = & y_1^1 \\
(14) & & + x_{25} & & = & y_2^1 \\
(15) & & & + x_{34} & = & y_3^1 \\
(16) & x_{14} & & + x_{34} & = & a_4 \\
(17) & \quad x_{15} & + x_{25} & & = & a_5
\end{array}\right\} \quad \text{(III)}
$$

$$
\left.\begin{array}{llllll}
(18) & x_{16} & & & = & y_1^2 \\
(19) & & + x_{27} & & = & y_2^2 \\
(20) & & & + x_{36} + x_{37} & = & y_3^2 \\
(21) & x_{16} & & + x_{36} & = & a_6 \\
(22) & & + x_{27} & \qquad + x_{37} & = & a_7
\end{array}\right\} \quad \text{(IV)}
$$

and $\qquad x_{ij} \geq 0$

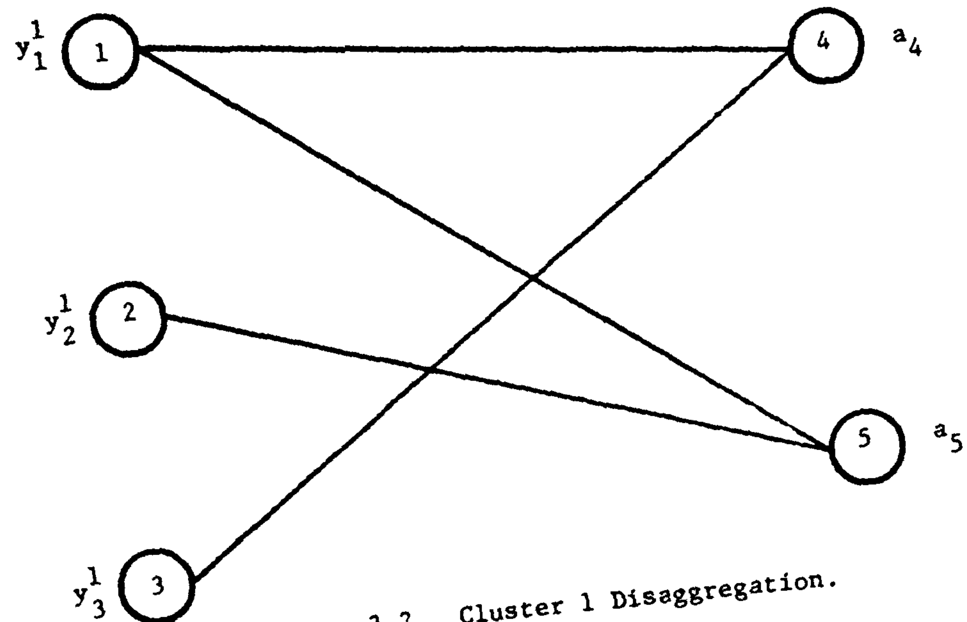


Figure 3.2. Cluster 1 Disaggregation.

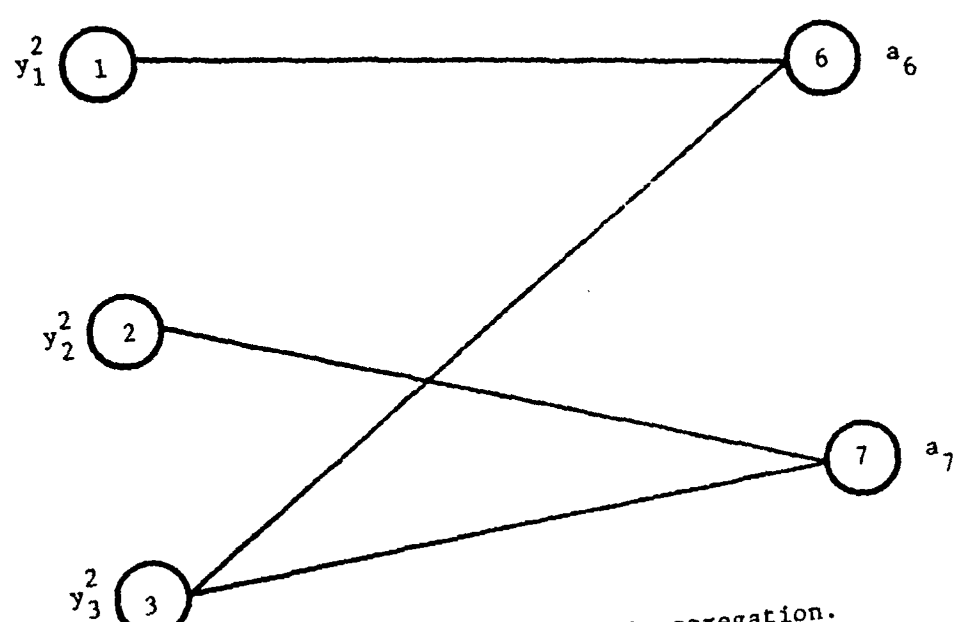


Figure 3.3. Cluster 2 Disaggregation.

4.0 THE ITERATIVE STEP

Once an aggregate problem has been solved and a feasible solution to the detailed problem generated, this information is used to decide how the aggregate problem can to be modified so that the procedure will move towards an optimal solution of the detailed problem. The goal is to add enough additional information to the aggregate problem to move it ultimately to an optimal solution of the detailed problem.

Methods have been developed to generate a dual variable solution to the detailed problem using the aggregate dual information. These dual variable values can be used to identify dual infeasible arcs in the detailed problem. (Dual infeasible arcs are those which do not satisfy the dual constraints.) Given dual infeasible arcs, various procedures have been suggested to reaggregate the detailed problem. Balas [1] suggests the following procedure. The aggregate problem duals are used to generate a dual variable solution to the detailed problem. All the nodes are separated into a cluster that have dual infeasible arcs incident to them. Singleton clusters are setup for each such node in the new aggregation. This procedure is iterated until the solution of the detailed problem is obtained.

The problem with this procedure is that the aggregate problem size tends to grow very rapidly. This may not be a desirable feature. Taylor [4] suggests a modified approach where only the most dual infeasible arc in each cluster is considered. The corresponding node to which it is incident is separated to form a single node cluster. This method uses the same motivation as the earlier approach but controls the growth of the aggregate problem size. The procedure uses the duals to identify the most dual infeasible arc in a cluster. From each cluster r,

the node $j_r \epsilon P_r$ is separated to form a cluster by itself in the reaggregated problem.

These procedures can be summarized as follows:

(1) The problem information is used to partition nodes into clusters of similar nodes.

(2) These nodes are aggregated to setup an aggregate problem

(3) The aggregate problem solution is used to generate upper and lower bounds on the original problem.

(4) The dual aggregate solution is used to generate a dual solution. This dual solution is used along with additional information about the problem instance, to reaggregate the problem.

(5) Steps (1) through (4) are repeated until an optimal solution to the detailed problem is obtained or until the user is satisfied with the sub-optimal solutions generated at some stage.

These procedures have two major shortcomings with regard to deployment planning problems. The first is that they only work for the situation where energy node in a cluster has the same connectivity to source nodes (i.e., if one sink and in the cluster is connected to a particular source node, then all sink nodes in the cluster are connected to the same source node). This condition is not normally satisfied for the MR assignment problem.

The second shortcoming is that a larger aggregate problem results after each iteration. If the computer system is constrained by storage space, then this can result ultimately in an unacceptably large problem.

The following reformulation has the potential to over come both of these problems.

5.0 MODEL REFORMULATION

For the example problem in section 2.0, consider the original constraints (1) through (7) and the new constraints (8) through (22). Substituting constraints (13),(14), and (15), constraint (11) can be rewritten as

$$x_{14} + x_{15} + x_{25} + x_{34} = a_4 + a_5$$

This is a relaxation of the original constraints (4) and (5). Similarly constraint (12) is a relaxation of constraints (6) and (7).

Constraint (8) together with constraints (13) and (18) provides

$$x_{14} + x_{15} + x_{16} = a_1$$

which is constraint (1). Constraint (9) with (14), and (19) provides (2); constraints (10) with (15) and (20) provides (3). Constraints (4) through (7) are identical to (16) through (17) and (21) through (22). Thus, if constraints (8) - (22) were compared to the original problem (1) through (7), the additional constraints are simply relaxations or can be used to generate the same original constraints. Constraint set (8) through (22) can be rearranged to form the system of equations

$$\text{Min} \quad \sum_i \sum_j c_{ij} x_{ij}$$

subject to: constraints (8) through (12)

and for cluster 1 : constraints (13) through (17)

for cluster 2 : constraints (18) through (22)

This system of equations has a matrix structure as shown in Figure 5.1. The matrix has a structure which consists of a master transportation problem linked via columns to subproblems which are also transportation problems.

5.1 General Model

The above specific example can be generalized with the following notation

Detailed Problem : $\quad \text{Min} \sum_i \sum_j c_{ij} \; x_{ij}$

$$\begin{array}{llllr} (23) & \text{s.t.} & \sum_j x_{ij} = a_i & i = 1\ldots M & \\ & & & & (VI) \\ (24) & & \sum_i x_{ij} = b_j & j = 1\ldots N & \\ & & x_{ij} \geq 0 & & \end{array}$$

Given a specified aggregation of detailed nodes an equivalent problem can be formulated as follows.

Let $r = 1,2,\ldots L$ be the number of clusters.

Let P_r denote the nodes j which comprise each cluster r.

Let y_i^r be defined as

(25) $\qquad y_i^r = \sum_{j \in P_r} x_{ij}$ for each $i = 1,2,\ldots M$ and $r = 1,2,\ldots L$ For each $r = 1,2,\ldots L$ add the redundant constraints

(26) $\qquad \sum_{i=1}^{M} y_i^r = \sum_{j \in P_r} b_j$

to problem (V). Constraints (26) are redundant since combined with the definitions of y_i^r in (25) a relaxation of the detailed problem constraints (24) is obtained. The y_i^r variable definitions (25) also result in an equivalent representation of the constraints (23) as

y_1^1	y_2^1	y_3^1	y_1^2	y_2^2	y_3^2	x_{14}	x_{15}	x_{25}	x_{34}	x_{16}	x_{27}	x_{36}	x_{37}	R.H.S.
1	1	1												$a_4 + a_5$
			1	1	1									$a_6 + a_7$
1			1											a_1
	1			1										a_2
		1			1									a_3
-1						1	1							
	-1							1						
		-1							1					
						1			1					a_4
							1	1						a_5
			-1							1				
				-1							1			
					-1							1	1	
										1		1		a_6
											1		1	a_7

Figure 5.1. Reformulated Matrix Structure.

$$(27) \quad \sum_{r=1}^{L} y_i^r = a_i \qquad i = 1,2,..M$$

The constraint sets (24) through (27) can be rearranged to obtain

$$\text{Min} \qquad \sum \sum c_{ij} x_{ij}$$

$$\text{subject to:} \quad \sum_{i=1}^{L} y_i^r = a_i \qquad i = 1,2,..M$$

$$\sum_{i=1}^{M} y_i^r = \sum_{j \epsilon P_r} b_j \quad r = 1,2,..L$$

(VI)

and for each cluster r, r = 1,2,..L

$$\sum_{j \epsilon P_r} x_{ij} = y_i^r \qquad i = 1,2,..M$$

$$\sum_{i=1}^{M} x_{ij} = b_j \qquad j \epsilon P_r$$

$$x_{ij}, \; y_i^r > 0$$

The matrix structure for the detailed problem is shown in Figure 5.2, and that for the reformulation is shown in Figure 5.3.

Every feasible set of x_{ij} for (VI) is also feasible to (V), and vice versa. This follows since only redundent constraints have been added to (V) in order to obtain (VI). Hence, solving (VI) also solves (V). A reformulation of the form (VI) can be constructed for any specified aggregation of nodes.

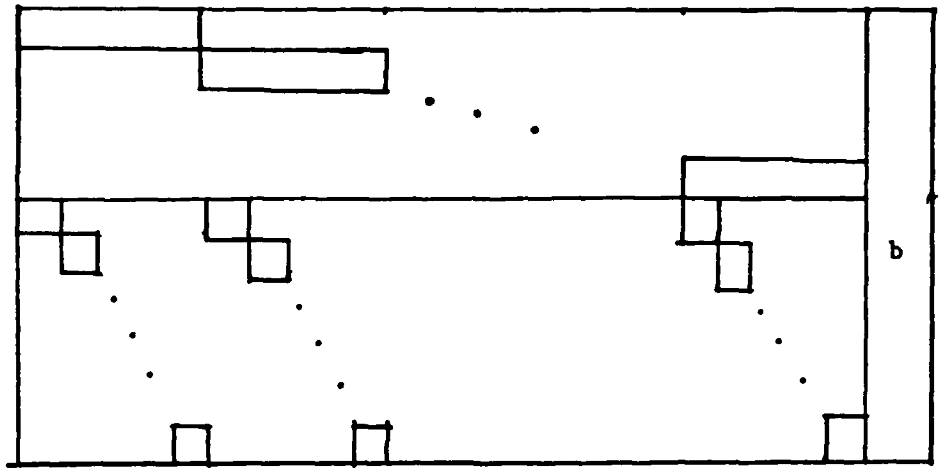


Figure 5.2. Matrix Structure of the Detailed Problem (V).

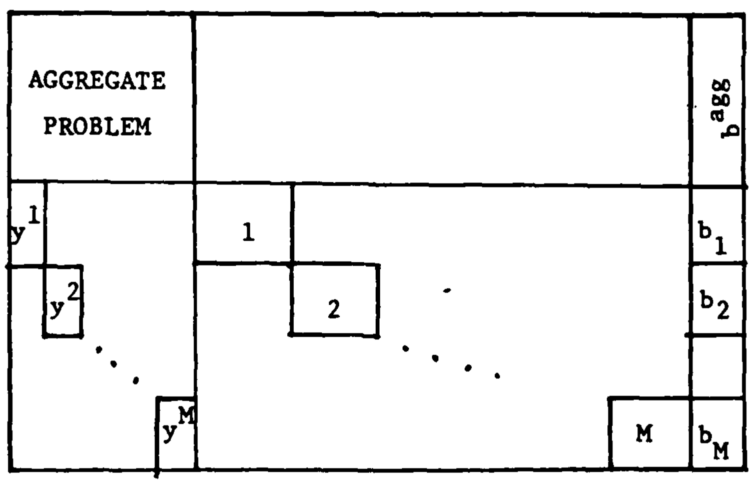


Figure 5.3. Matrix Structure of the Reformulated Problem.

5.2 Solution of the Reformulated problem

There are two extreme cases with regard to the reformulated problem. If each node forms a cluster then the aggregate problem is as large as the original problem. If all nodes together form a cluster then the aggregate problem is trivial. In the latter case there is just one arc out of each source node to the aggregate sink node, and the solution is to fill the arc to the source supply value. Between these two extremes, there is a distribution of time and complexity of solution. These vary between the aggregate and detailed levels which allows the problem to be solved within the time and space specifications. Considerable latitude is available in trading-off size the aggregate problem versus size and number of the disaggregated problems.

An immediate approach to solving the reformulated problem is to setup the aggregate problem as a master problem. The subproblems are formed by the disaggregation problems linked together by Benders' constraints [2]. For the problem discussed in section 5.0, the master problem would be

$$\text{Min} \qquad z$$

$$\text{subject to:} \quad \sum_{r=1}^{2} y_i^r = a_i \qquad i=1...3$$

$$\sum_{i=1}^{3} y_i^1 = a_4 + a_5$$

$$\sum_{i=1}^{3} y_i^2 = a_6 + a_7$$

one Benders' constraint for each iteration and

$$y_i^r > 0 \qquad \text{for } i=1...2, \quad r=1...2$$

The subproblems for a given y_1^r would be

SUBPROBLEM 1

$$
\begin{array}{llllll}
\text{Min} & c_{14}x_{14} + c_{15}x_{15} + c_{25}x_{25} + c_{34}x_{34} & & & & \\
& x_{14} + x_{15} & & & = & y_1^1 \\
& & x_{25} & & = & y_2^1 \\
& & & x_{34} & = & y_3^1 \\
& x_{14} & & x_{34} & = & a_4 \\
& & x_{15} + x_{25} & & = & a_5
\end{array}
$$

SUBPROBLEM 2

$$
\begin{array}{llllll}
\text{Min} & c_{16}x_{16} + c_{27}x_{27} + c_{36}x_{36} + c_{37}x_{37} & & & & \\
& x_{16} & & & = & y_1^2 \\
& & x_{27} & & = & y_2^2 \\
& & & x_{36} + x_{37} & = & y_3^2 \\
& x_{16} & & + x_{36} & = & a_6 \\
& & x_{27} & + x_{37} & = & a_7 \\
& & & \text{and} \quad x_{ij} & > & 0
\end{array}
$$

Benders' constraints of the form

$$z > \sum_i u_i^1 y_i^1 + \sum_i u_i^2 y_i^2 + v_1^1 a_4 + v_2^1 a_5 + v_1^2 a_6 + v_2^2 a_7$$

are added to the master problem at each iteration, with the u's and v's being the duals of subproblems 1 and 2.

Benders' procedure is discussed in detail in PDRC reports 84-09 and 85-03. The procedure involves solving the master problem (in this case the aggregate problem) and then solving the subproblems (in this case the

disaggregation problem). Based on the solutions to the subproblems, a Benders' constraint is constructed and added to the master problem. The new master problem is solved and the process repeated. A lower and upper bound on the value of the optimal objective function of original detailed problem is generated at each iteration. When the two bounds are equal, the process terminates with an optimal solution.

5.3 Advantages and Disadvantages of the Benders' Approach

Benders' decomposition is suggested to solve the reformulated problems. The basic advantage of Benders' procedure is that it separates the solution of the aggregate problem and the disaggregate subproblems. Once the aggregate problem is solved, the disaggregation subproblems, one for each cluster, can be solved independently. The linking Benders' constraints generated by the subproblem, one constraint for each cluster, are added to the aggregate problem, guiding it towards the solution of the entire problem. Furthermore, at each iteration upper and lower bounds on the optimal objective function value are generated. These bounds permit termination when the solution is reasonably close to optimal.

A problem with the use of Benders' decomposition is that the constraints added to the aggregate problem cause it to be a network flow with side constraints model. This reduces the advantage of its original pure network structure. Research is in progress on a procedure which permits the additional Benders' constraints to be used to change flow capacities in the aggregate transportation model. This would allow use of network algorithms to solve the aggregate problem. Based on

information available at an iteration, the aggregation itself may also be changed to hasten the solution.

While Benders' decomposition is a procedure which can be used to solve the reformulated model, other procedures which exploit the network structure of the aggregate and disaggregation problems many expedite the solution mechanism for the reformulated model.

6.0 EXTENSIONS AND RESEARCH AREAS

The concepts discussed in this report suggest various areas for future research.

(1) Though the aggregation has been discussed with respect to a two level solution procedure, the procedure could be extended to a multi-level hierarchical procedure. The decisions are refined at each level with preceding level providing resource constraints for succeeding levels.

(2) Since any minimum cost flow problem on a general network can be transformed into a transportation problem as in Lawler [3], the aggregation procedure could be applied to solve minimal cost flow problems. This aggregation and problem reformulation might be carried out without transforming the network, thereby working directly with the network itself.

(3) Since the master and subproblems are transportation problems, techniques which replace the Bender's cuts with equivalent capacities on aggregate arcs might be devised so that the master problem remains a network flow problem, rather than become a network flow with side constraints model.

(4) In solving the master problem, the Benders' might retained in the objective function. This would produce a network flow problem with a minimax objective function. In this case the objective function is a piecewise linear convex objective function.

(5) Even though only sink node aggregation has been discussed, it can be extended to aggregation of both source and sink nodes by performing the operation sequentially.

References

1. Balas, E., "Solution of Large Scale Transportation Problems through Aggregation," Operations Research 13: 82-93 (1965).

2. Lasdon, L.S., "Optimization Theory for Large Systems," Macmillan (1970).

3. Lawler, E., "Combinatorial Optimization: Networks and Matroids," Holt, Rinehart and Winston (1976).

4. Taylor, R.W., "Aggregate Programming in Large Scale Linear Systems," Ph.D. dissertation, Georgia Institute of Technology (1983).

5. Zipkin, P.H., "Aggregation in Linear Programming", Ph.D. dissertations, School of Organization and Management, Yale University (1977).

END

DTIC

6-86